Lab 4A – Dice Game

You will create a game involving two six-sided dice.

Players start with 500 points (or any point amount you wish to use).

Before each roll of the dice, players must risk points according to these rules:

- must risk at least 1 point
- can't risk more points than the points they currently have

When the dice are rolled…

- their images are displayed
- the total value of the dice is displayed

Points are added and subtracted according to these rules:

- total of the dice greater than 7 – gain the points risked
- total of the dice equals 7 – no points gained or lost
- total of the dice less than 7 – lose the points risked

The display of total points should update after each dice roll.

Doubles are also significant in this game. There is a limit on how many doubles players can roll in this game – when they reach that limit, the game is over. Since this limit is a fixed value, it should be declared as a constant. In the provided example, the doubles limit is 10 and players can keep their points when they reach the limit (they don't lose, but they are done playing).

After each roll of the dice, when users…

- roll doubles, they should see some kind of notification (probably in a label, but you could use a message box)
- don't roll doubles, they should not see any notification

The game also ends when users have 0 points (users should never end up with negative points as they are not allowed to risk more than the total points they currently have).

So, the game ends when…

- players have 0 points
- players have reached the doubles limit

When the game ends…

- notify users that the game is over and why it has ended (could be a message box or a label – the sample program uses a message box)
- disable the ROLL DICE button so users can see the final results.

Players should have access to the instructions (with a button). If you make alterations to the rules (setting a target point value that makes you a "winner" if you reach it, for example, or adding more to the significance of doubles), that is OK – make sure these are explained in the instructions so players (especially the instructor!) understand the rules of the game.

Objectives

- Evaluate decisions and conditions
    - If…Then statement
    - If…Then…Else statement
    - If…Then…ElseIf statement
- Use of relational & logical operators
- Exception handling with try-catch
- Use of random numbers
- Use of running totals and counters

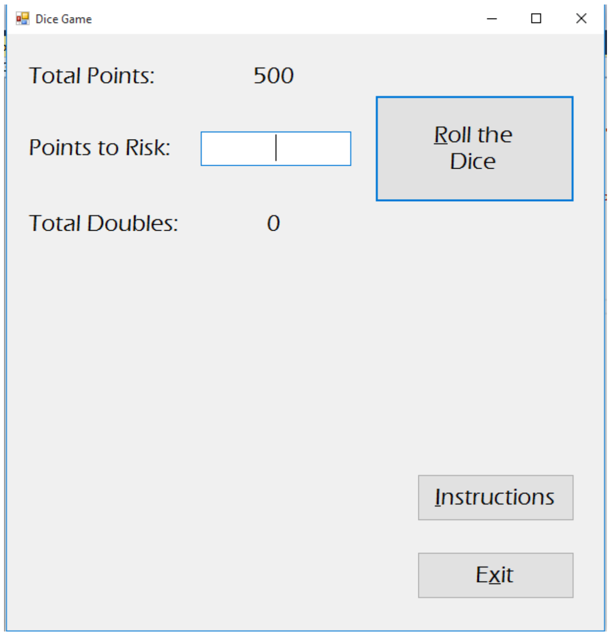
Requirements

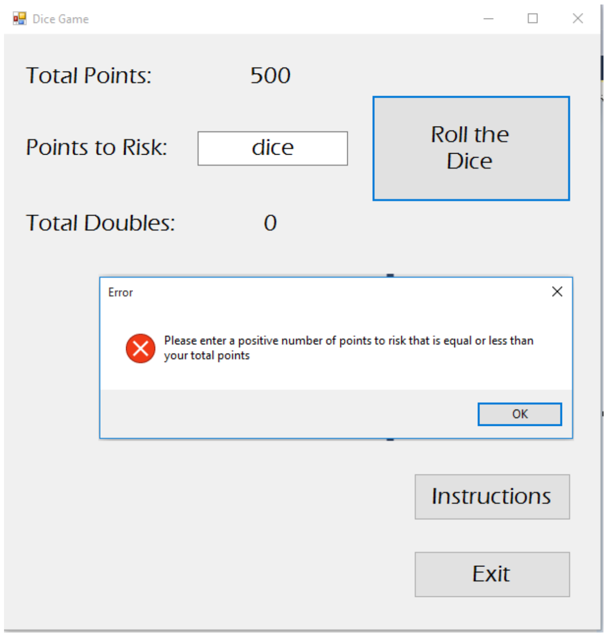
- Comment your code!!!
- Follow the details given on pages 1-2. Many of the situations are addressed with If, If-Else, or If-ElseIf.
- Display the instructions in a message box when players click an INSTRUCTIONS button. You can get a new line by using `vbCrLf`
  `"Line 1" & vbCrLf & "Line 2"` or `"Line 1" + vbCrLf + "Line 2"`
- You can display an image in a PictureBox with code similar to this:
  ```
  picDie1.Image = My.Resources.die1
  ```
- Users enter the points to risk in a TextBox.
- The limit for doubles is 10 (or any value you choose). This fixed value should be declared as a constant.
- Declare your Random object as a class-level variable – then it is only "seeded' once.
- Handle the exception if a numeric value is not entered for points risked.
- Use variables for any data you are collecting and displaying. You may find variables useful for some calculation work performed during the program. Consider what data types make sense for the kind of values you plan to store in the variables.
- Here's something new – you are not required to use it, but you might want to try!
  - You want to keep the values of total points and total doubles (and not lose them) every time players roll the dice. One way to do this is to declare your variables for total points and total doubles as class-level variables (this is the only way you've learned so far that would work) using the keyword `Dim`.
  - Another way is to declare your variable in the exactly one sub where it appears using the keyword `Static`. It could look like this:

    ```
    Static intAllThePointsYouHave As Integer = 500  'A silly variable name
    ```

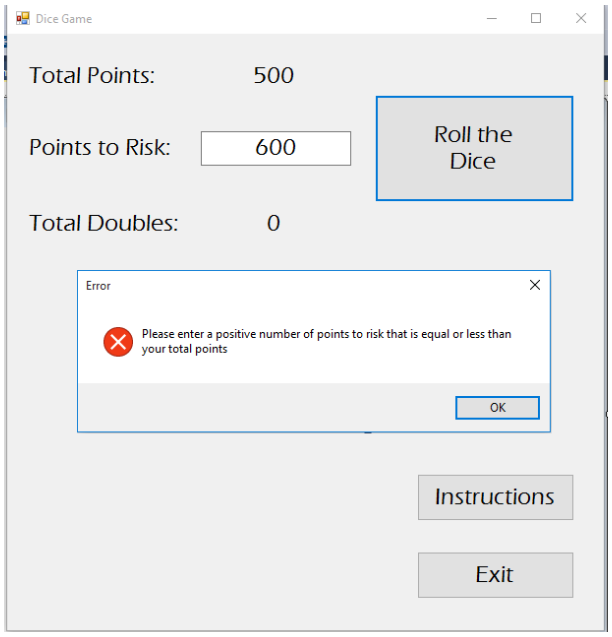
  - The value will be set to 500 only the first time that line of code executes and the variable will keep/remember its value every time the sub for clicking on the ROLL DICE button executes.
- ROLL DICE (or whatever text you use) is the form's AcceptButton; a CancelButton is not necessary
- Tab order: points to risk text box, ROLL DICE button, INSTRUCTIONS button, and EXIT button
- Include keyboard access to ROLL DICE, INSTRUCTIONS, and EXIT buttons
- After you have informed players that something is wrong with the data in the textbox, assign the focus to the textbox so they can take care of it without having to click in the textbox themselves
- Type and modify the following at the top of the code:
  ```
  'Your Name
   'Date
   'Assignment Name (Lab 4A – Dice Game, for example)
   'Computer Programming I - Bower (or CS 11400 – Bower if you want IPFW class)
   'This program will... (describe the program)
  ```

Sample Data (remember that the access key underscores only appear when users press ALT – they are only displayed in the first image)





Please enter a positive number of points to risk that is equal or less than your total points



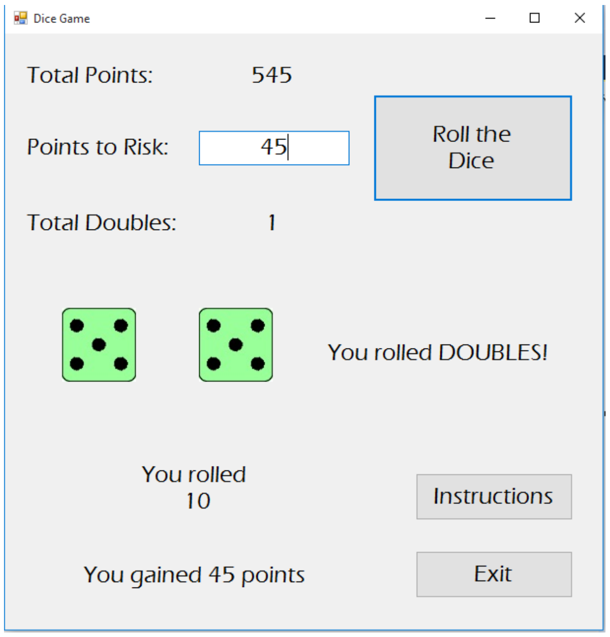
Total Points: 500

Points to Risk: 600

Total Doubles: 0

Error

Please enter a positive number of points to risk that is equal or less than your total points



Total Points: 545

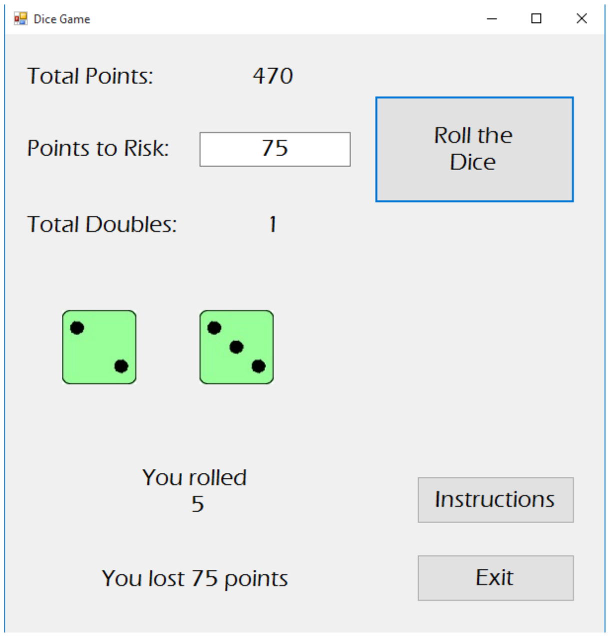
Points to Risk: 45

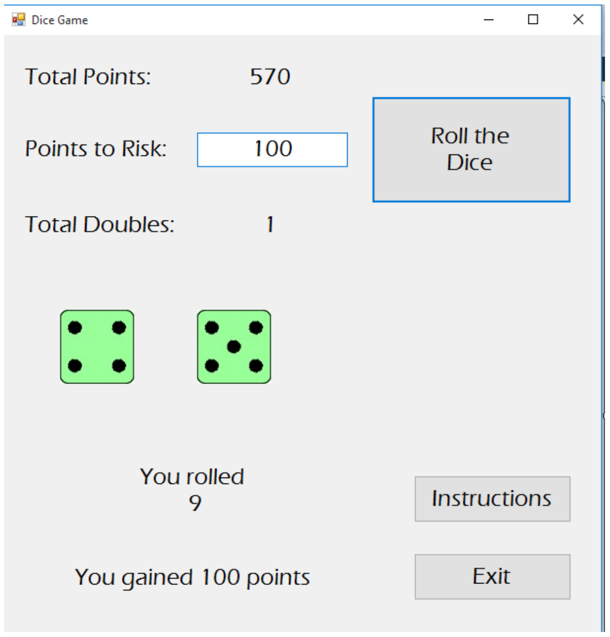
Total Doubles: 1

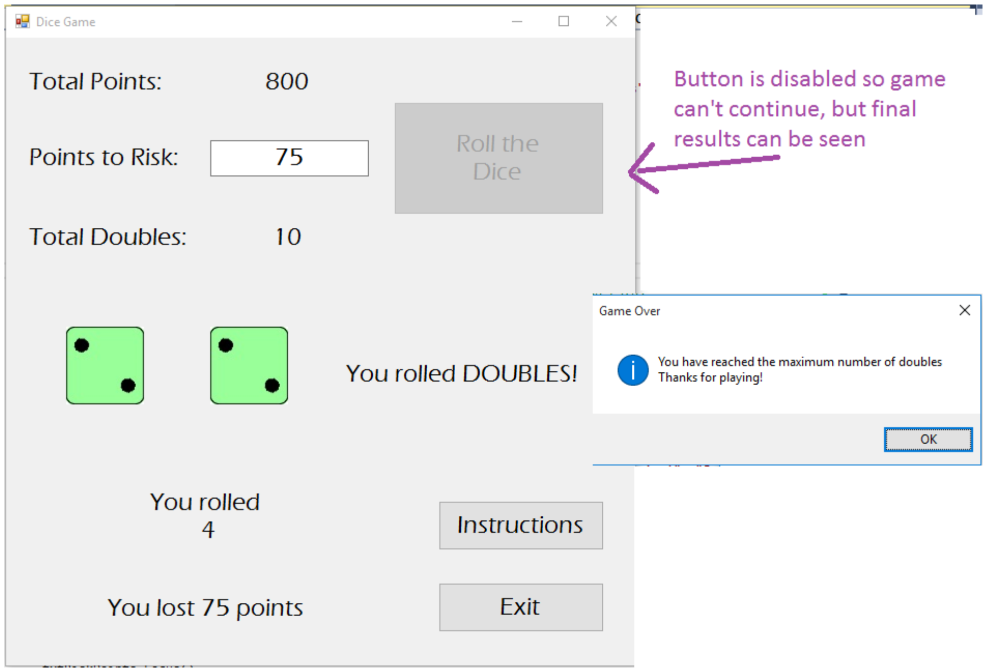
You rolled DOUBLES!

You rolled 10

You gained 45 points


Dice Game
Total Points:
470
Points to Risk:
75
Roll the Dice
Total Doubles:
1
You rolled
5
Instructions
You lost 75 points
Exit


Dice Game
Total Points:
570
Points to Risk:
100
Roll the Dice
Total Doubles:
1
You rolled
9
Instructions
You gained 100 points
Exit



Dice Game

Total Points: 800

Points to Risk: 75

Total Doubles: 10

You rolled DOUBLES!

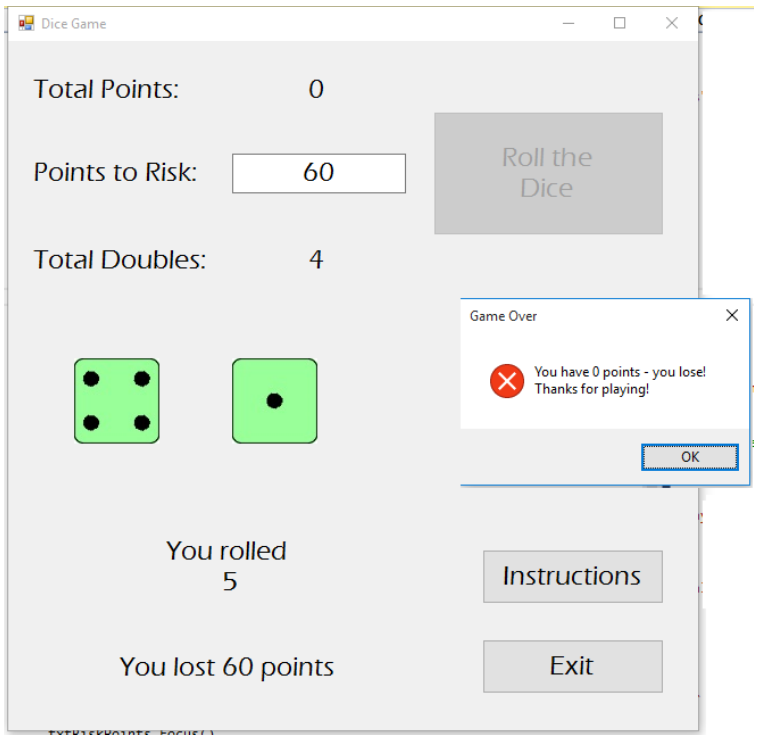
You rolled
4

You lost 75 points

Game Over

You have reached the maximum number of doubles
Thanks for playing!

Roll the Dice
Instructions
Exit
OK
Button is disabled so game can't continue, but final results can be seen



Dice Game

Total Points: 0

Points to Risk: 60

Total Doubles: 4

You rolled
5

You lost 60 points

Game Over

You have 0 points - you lose!
Thanks for playing!

Grading

Form design / Control names	3
Runs properly	2
Comments/documentation included	3
Accurate results (points added/subtracted accurately)	4
Variables & constants names declared properly	2
Exit button	1
Instructions button	2
Roll Dice button (displays dice and dice total)	2
Exception handling (catches unacceptable input)	2
Run-time communication	2
Random number generation	1
Game ends correctly	2
Doubles (notification and keeping a counter of # of doubles)	2
Check for valid user risk amount	2
TOTAL	**30**